SPATIAL DOMAIN IMAGE STEGANOGRAPHY

by

K. DURGA PRASAD

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

KANPUR

MARCH, 2002

SPATIAL DOMAIN
IMAGE STEGANOGRAPHY

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

DIIT

by

K. DURGA PRASAD

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

KANPUR

MARCH, 2002

22-3-02

CERTIFICATE

It is certified that the work contained in the thesis entitled ***"Spatial Domain Image Steganpgraphy"*** by **K Durga Prasad**, (Roll No. Y022403) has been carried out under my supervision, and this work has not been submitted elsewhere for the award of a degree.

March, 2002

S Venkatesh

K.S.Venkatesh
Assistant Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur.

ABSTRACT

Steganography is an art of secret communication. In this thesis we have tried to achieve spatial domain image steganography, by the method of image differencing, in which a secret image can be hidden into another image with much imperception. An attempt has been made to achieve steganography for grey scale and color images. This thesis also studied the fidelity and capacity considerations of spatial-domain steganography. This method is also applied to achieve video steganography.

Dedicated to

DOORDARSHAN

Acknowledgements

At the very first I am thankful to the almighty God, who provided me everything I needed to complete the course.

*It is a great pleasure to express my deep sense of gratitude to **Dr.K.S.Venkatesh**, my thesis supervisor for his invaluable guidance, constant encouragement and an uncanny knack of solving any problem systematically.*

I owe special thanks to Dr.Sumana Gupta for teaching the basics of Image Processing which helped me greatly during the thesis work.

I am extremely grateful to the Directorate General of Doordarshan for giving me an opportunity to undergo this course at this prestigious Institution. I extend my sincere thanks to seniors Ramana, Janardhan, Gangadhar and Harsha. I always remember the moments spent with my classmates.

Now I am paying special thanks to Mr.Yeshvant Kanitkar and Mr.Sanjay Khirwadkar of Nagpur who made my life easier with 'C' and during whose interaction I was familiarized with text hiding (Cryptography). Finally, thanks to my parents and my wife, Mouli for tremendous & cheerful support.

-K.Durga Prasad

CONTENTS

CHAPTER 1

Introduction

1.1 INTRODUCTION

Steganography is an art of secret communication. Its purpose is to hide data and the data can be anything: digital images, sound files, text files, etc. One can use images, sound files, and other computer files that contain perceptually irrelevant or redundant information as covers or carriers to hide secret messages. After hiding secret data into the cover, we obtain a so-called stego, which should not contain any detectable artifacts due to message embedding. The less information one embeds into the cover, the smaller is probability of introducing detectable artifacts by the embedding process. The requirements of any data hiding technique are:

1. The stego message should not reveal the existence of the secret message.
2. The embedding capacity should be as high as possible.
3. Only the receiver should know the existence of the secret message and he alone should be able to recover the secret message.
4. The content of the secret message should not be distorted during transmission. i.e. embedded message must be robust to noise.

These are the main features, which characterize any data hiding. Steganography is a hiding scheme that hides the secret data and even the very presence of secret data. Other techniques of data hiding are cryptography, watermarking, etc. Cryptography aims to make the communication unintelligible to those who don't possess the right keys, but makes no attempt to hide the existence of the data itself. Watermarking, which is similar

to steganography, hides only small amounts of data, used for copyright protection: here the emphasis is again on hiding the watermark as well as its existence, while causing the least possible distortion of the watermarked image.

Steganography finds immense applications that include the following areas:

1. Military and Intelligence communication.
2. Copyright protection, Access control.
3. Authentication and tampering detection.
4. Video broadcasting, where an additional signal can be sent on the same channel along with main signal, saving bandwidth considerably.

Steganography can be carried in spatial domain as well as in frequency domain. In spatial domain secret message is embedded in intensity of the cover whereas in frequency domain only small quantity of message is embedded in transform coefficients with high robustness. Frequency domain methods include DCT, DWT, randomly sequenced PPM, secure spread spectrum methods, etc. In spatial domain methods, the most common and easiest method is LSB insertion method in which two or more bits of least significant bits are used for embedding. In this method one can achieve imperceptible embedded communication to casual views but stegoanalysts can use statistical properties to detect the existence of the secret message. Moreover, the stego fidelity degrades if more bits are used.

1.2 Motivation for the problem

Data hiding is a rapidly growing field with potential applications in areas mentioned in the above section. In cryptology, for example, the messages are encrypted, after which only the permitted user can see meaningful communication. All these are perceptual data hiding techniques that reveal the existence of a secret message. If secret data is embedded into the covers, which contain perceptually irrelevant information, the existence of the secret message is not revealed. If images are used as a cover, higher embedding capacity can be achieved easily by exploiting certain properties relating to the human visual system (HVS). This motivated me to turn to image steganography.

1.3 Organization of the Dissertation

This thesis is organized into a total of 5 chapters. In chapter 2, the properties of HVS, which enable us to hide secret images, and the steps involved in image hiding are introduced. Also emphasized in this chapter are image differencing, quantization, grey value adjustment algorithms and algorithm performance evaluation criteria. In chapter 3, secret image embedding and making of the stego are discussed. In chapters 4, secret image recovery and the application of the same steganography methods to color images and image sequences are discussed. In chapter 5, our conclusions and suggestions for future work are presented. All the results obtained are attached at the end of each respective chapter.

CHAPTER 2

Image differencing, Quantization and Grey value adjustment

2.1 Introduction

In this thesis, a method to embed a grey valued secret image into grey valued cover image is made possible by exploiting the following two properties:

1. High spatial correlation between neighboring pixels of any image i.e. similarity among grey values of consecutive image pixels.
2. Human visual system's (HVS) varying insensitivity over smooth and contrast-filled regions of a picture: the HVS is less sensitive to changes made in high contrast regions of the image.

The fundamental principle that underlies image hiding in the spatial domain is to replace the image pixels of the so-called 'cover' image with the pixels of the 'secret' image, taking care that the replacement pixel values match as closely as possible, the values of the pixels they replace. This ensures that the cover image, thus modified, still closely resembles the original. A careful record is kept of the replacements carried out, so that the secret image can later be 'reassembled', so to speak, from the modified cover (stego) image. It must be understood, that, in this process, the secret image is recovered *exactly*, what suffers distortion is only the cover image. Our study will aim towards keeping this distortion to the minimum. Furthermore, we traverse through the cover image in an order provided by a pseudorandom sequence in order to achieve secrecy, and prevent illicit users from tampering with/accessing the hidden data.

The histograms of different images are in general, very different in shape. But if we perform image differencing between every pixel and its adjacent pixel, using the above first property, the histograms get very similar shapes. This fails only in computer-generated (artificial) images. This similarity allows us a way to embed a secret image easily in a cover image. The second property helps us to embed more information in contrast areas and less in smooth areas as the pixel value modification to cause just noticeable change in contrastive areas is much more. Edge and contrast areas can suffer considerable changes without perception. This helps to hide more data imperceptibly in an image.

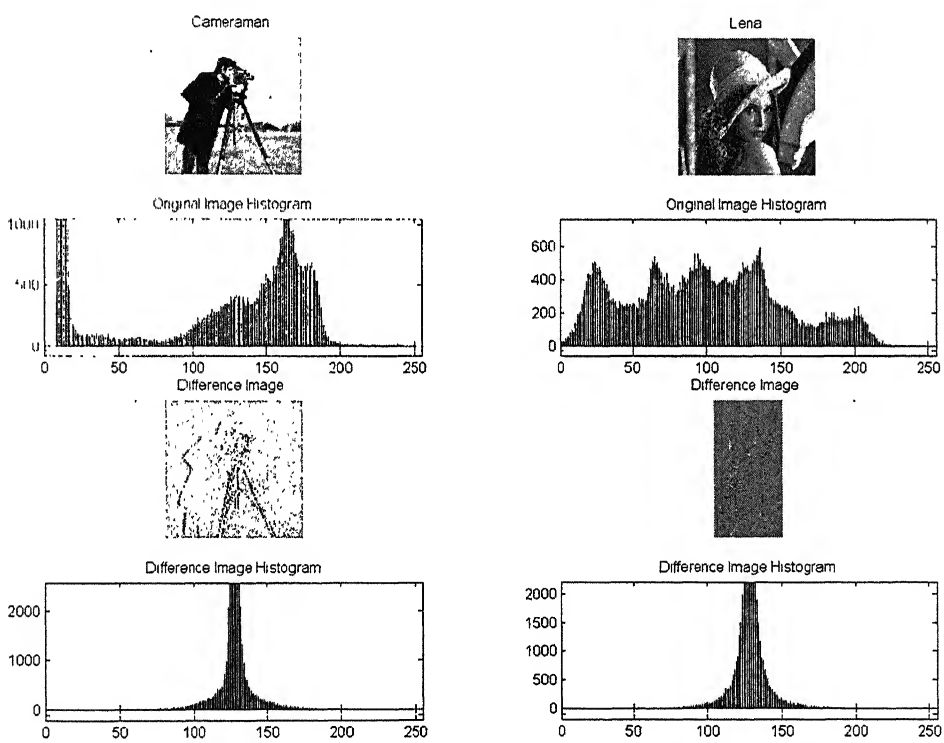


Fig 1. Histograms of original and difference images

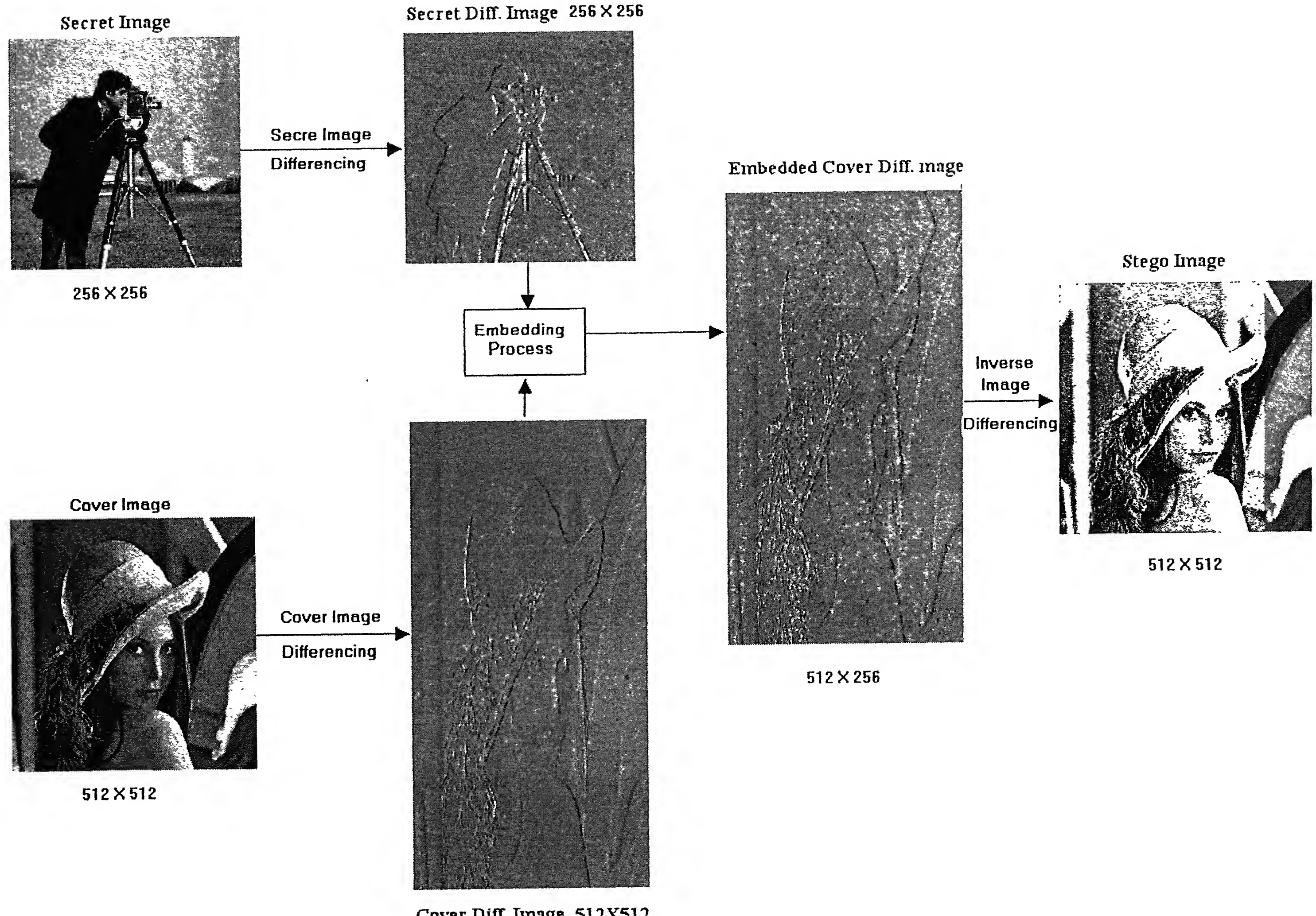


Fig 1. Image hiding using image differencing

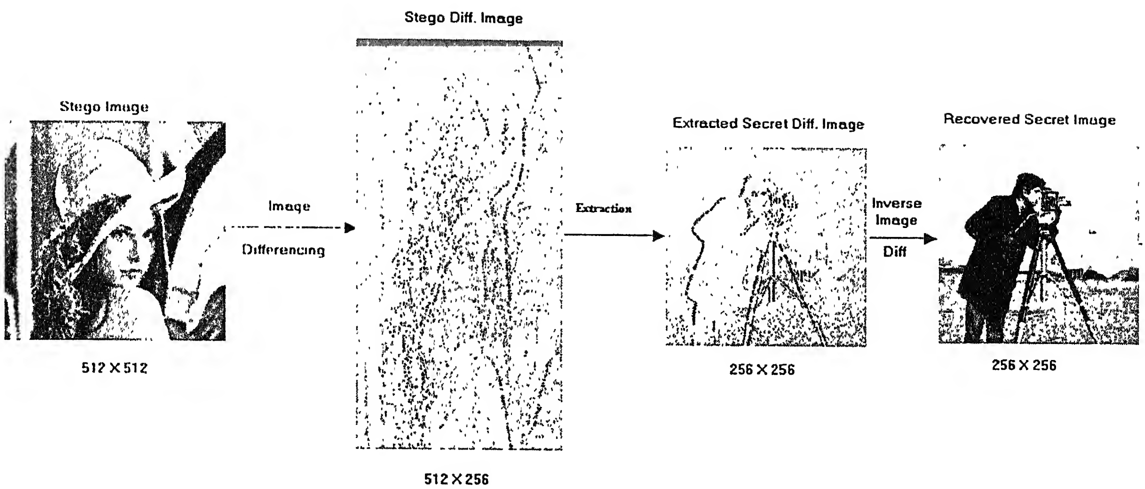


Fig 2. Reconstruction of Secret Image

The important steps involved in this method of image hiding are:

1. Perform image differencing on the secret image and the cover image to get the et difference image and cover difference image respectively
2. Quantization of the difference grey values of the secret difference image and cover difference images based on considerations of HVS peculiarities.
3. Grey value adjustment of the cover difference image to enable grey value replacement.
4. Replacement of the grey values of the cover difference image with that of the secret difference image in an order determined by a pseudorandom sequence to get the stego difference image.
5. Perform inverse differencing on the stego difference image to get the stego image.

2.2 Secret and Cover Image differencing

Image differencing is achieved by taking the difference of adjacent pixel intensities, while traversing through the image. The same traversing order has to be followed when inverse differencing has to be performed. The difference can be taken between adjacent pixels that are overlapping and also between non-overlapping pixels, taking advantage spatial correlation among adjacent pixels. In former kind of differencing, the resultant image is of the same size as that of the original image, whereas in the latter case it would be half that of the original.

The former method of differencing is performed on the secret image to obtain the secret difference image and the latter on the cover image. The primary advantage of non-overlapping differencing is that it is more robust to noise. The inverse differencing procedure for this case does not allow accidental errors to propagate indefinitely: each consecutive pixel pair (as well as any errors that may occur in it) is isolated. This property recommends its use for the cover image, so as to make the stego more robust to noise during transmission. This has been explained clearly in chapter 3. The price to be paid is that the effective size of the (differenced) cover, and therefore its hiding capacity, is reduced to half, so that an embedding capacity of only upto 50% of the size of the cover image, can be achieved with imperceptions.

In the secret image, each pixel's intensity is subtracted from that of the adjacent pixel. Let S be the secret image. We number its pixels in zigzag order starting from the first row: traverse the first row from left to right, the second row from right to left, the third again from left to right, and so on. Let S(i) be the pixel intensity of the i^{th} pixel taken in S in this order. S_d shall be the secret difference image and $S_d(i)$ be the i^{th} pixel's intensity in S_d. Image differencing is performed in Zigzag order to obtain S_d from S through the equation:

$$S_d(i) = S(i) - S(i-1) + 128$$

Where $i = 1,2,3...n.$; n = total number of pixels in the S image.

For the sake of convenience, we assume $S(-1) = 0$ so that $S_d(1) = 128$. Thus the secret difference image produced is of the same size as that of the secret image.

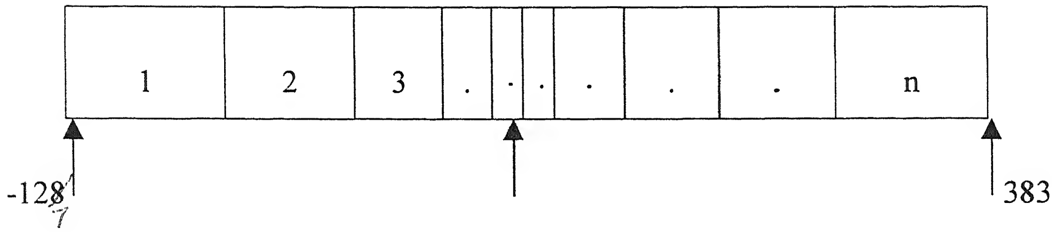
In the cover image, the differencing is performed between pairs of adjacent non-overlapping pixels. Let C be the cover image and C_d be the cover difference image. Image differencing is performed in zigzag order (re. above) through the equation:

$$C_d(i) = C(i) - C(i-1) + 128$$

where $i = 2,4...n.$; n = total number of pixels in the C image. Thus the size of the cover difference image is half that of cover image –rows in C_d are half the width of the rows in C.

2.3 Quantization:

After getting the secret and cover difference image values, the next step is the quantization of these values into 'n' ranges indexing these ranges from 1 through 'n'. A difference value, which falls in the range 'k', is said to have index 'k'. All the values of a certain range are considered as close enough to substitute for one another, and if one value is replaced with another in the same range in the original image, the change, one holds, cannot be easily perceived. It stands true when image differencing is performed between non-overlapping adjacent pixels. These ranges are used to get exchange values among them, in the cover difference image.



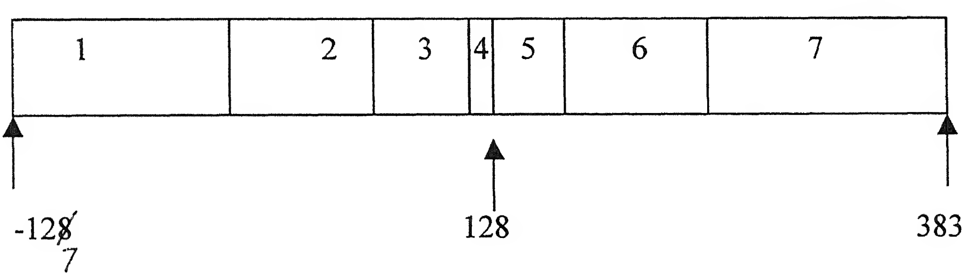
The range intervals of the difference values are chosen based on the human visual system's variation sensitivity from smooth to contrastive areas. The pixels in the contrastive areas are with difference values far from 128(as we added 128 to the each difference value) and these may tolerate larger changes when compared with smoother areas, which are with difference values close to 128,when judged with the same sensitivity by a human observer. This suggests that we create larger range sizes at places far removed from the intensity level 128 and smaller ones in the vicinity of 128 for the purpose of better replacement (replacement resulting in less noticeable changes). This strategy helps to embed more information in the contrastive areas with less perception of distortion.

Necessity for Adjustment:

In the embedding process, for each pixel p_{sd} in the secret difference image S_d, we find a pixel p_{cd} in the cover difference image C_d whose gray value is the same as that of p_{sd} and hide p_{sd} in its place by replacing the value of p_{cd} with that of p_{sd}. Although the grey value distributions of S_d and C_d are similar, there may exist an insufficient number of pixels in C_d in certain ranges, which have same (quantization) index as that of certain pixels in S_d which one wishes to embed. So there arises the need for the grey value adjustment before image embedding in the cover difference image of some pixels into new values, which are insufficient for embedding.

This grey value adjustment procedure is the crux of the whole embedding process, as this will directly affect the Peak Signal to Noise Ratio (PSNR) of the embedded image. The adjustment work starts by counting the total number of pixels in each range (index) of the secret difference image S_d and cover difference image C_d respectively. The counts of indices are then checked to find out deficit indices. We say index 'k' is deficit if total count (size) of index 'k' in image C_d is smaller than that of image S_d. Ranges that are not deficit are called as excessive. For every deficit range we have to accumulate enough pixels in C_d that are not with index 'k' to compliment the amount of insufficiency. This is done by changing the values of (deficit amount) pixel intensities of the excessive range to any value of the deficit range – effecting a migration. The accumulation work will be considered as complete when the deficit range collects sufficient numbers of pixels. There may exist more than one deficit range, which need be processed. The order of processing deficit ranges also affects the quality of the stego image.

Here depending upon the order of the processing of more than one deficit range, different algorithms are proposed. Different algorithms lead to different PSNR values of the stego image. In the study of these, Algorithm five is found to be the best one.



Ranges 3,4, and 5 are considered smoother than ranges 1,2,6 and 7 as they contain difference values closer to 128, an indication that they represent smoother areas of the original cover image. The occurrence of deficit ranges is clearly affected also by the choice of range boundaries in the first place, as well as by the number of ranges we partition the difference histogram into.

The aim of any of any algorithm should be to select pixels from excessive ranges for a minimum distance transfer to a deficit range and at the same time one has to note that pixel transfer for a high contrast deficit range (distant from 128) can tolerate more distant transfers in comparison to that for smoother deficit range.

Based on the above conclusion, we have defined a cost factor, which is a function of the adjustment algorithm applied, the number of ranges and the boundary levels of the ranges. Different range boundaries can be exercised to get minimum cost factor, which results in better PSNR of the stego image. The greater the number of ranges, lesser is the cost. A greater number of ranges require more bits to represent the range index number. The formula for the cost factor is given below.

$$R = \sum \sum \mid m_k - m_l \mid * n_{kl}$$

$$k, l = 1 \text{ to No. of ranges.}$$

Where R = cost factor.

m_k = conditional mean of the range 'k'.

m_l = conditional mean of the range 'l' .

n_{kl} = No of pixels transformed from 'l' excessive range to 'k' deficit range.

$R \ \alpha \ 1/\text{No. of ranges.}$

So a trade-off between the number of ranges (Bits to represent) and cost factor (PSNR of the stego) has to be achieved. If the boundary ranges are carefully exercised for a give number of ranges, minimum R can be achieved.

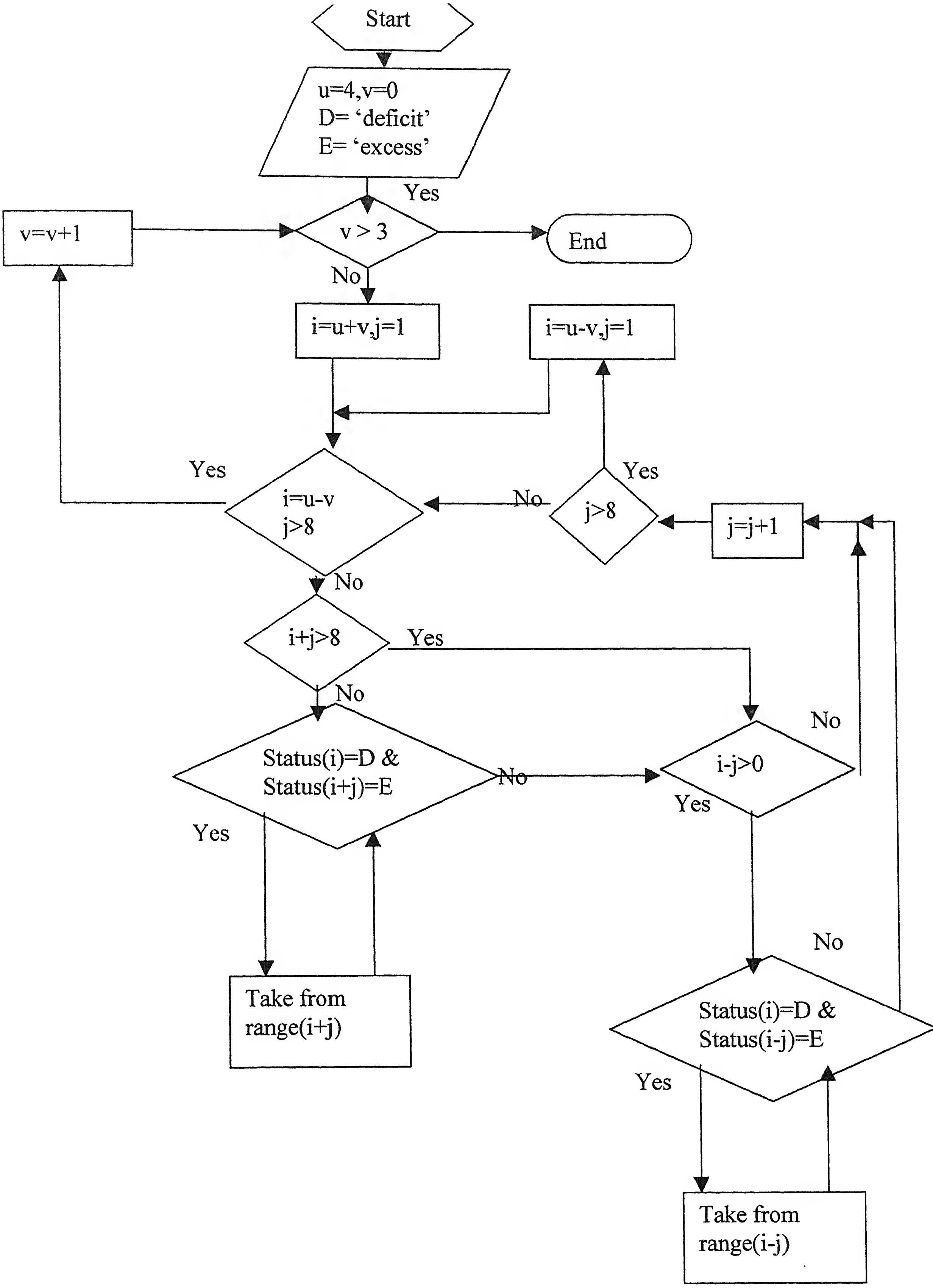
Also when the number of ranges has to be increased for a given secret image and cover image, we take the boundary at the median of a certain range, which enables us to get fewer deficit pixels and a lower cost factor. Otherwise small n_{kl} and R cannot be guaranteed.

2.4 Discussion on Range Boundary Selection and Adjustment Algorithms

The range intervals of the difference values are chosen based on the human visual system's variation sensitivity from smooth to contrastive areas. The pixels in the contrastive areas are with difference values very far from 128 (as we added 128 to each difference value) and these may tolerate larger distortion changes without perception when compared with smoother areas, which are with difference values close to 128, when judged with the same sensitivity by a human observer. This suggests that we create larger range sizes at places far removed from the intensity level 128 and smaller ones in the vicinity of 128 for the purpose of better replacement (replacement resulting in less noticeable changes). This strategy helps us to embed more information in the contrastive areas with less perception of distortion. It is to be noted that a PSNR change of 0.5 dB is visible to human observer. This was discussed earlier.

The necessity of grey value adjustment arises due to improper selection of boundaries for the ranges and the dependence of boundary position upon the secret image histogram. Hence for each secret image, if appropriate range boundaries are chosen, it would give better PSNR even if grey value adjustment is required. Here we mention one approach for selecting range boundaries.

Select the number of ranges the difference histogram is to be divided into, $K=2^n$ for some integer n (n = number of bits needed to refer to a particular range index). Starting from the middle (i.e., 128) and going away in either direction (left and right), collect a minimum number of pixels equal to [total pixels/No. of ranges]. Each time the number of ranges needs to be raised, repeat the same procedure. But this may not ensure an increase of the PSNR from the previous case. To ensure that the PSNR improves, we need to set a new boundary in the largest range at its median. This ensures that PSNR increases, cost factor decreases and total number of deficit grey values remains constant.



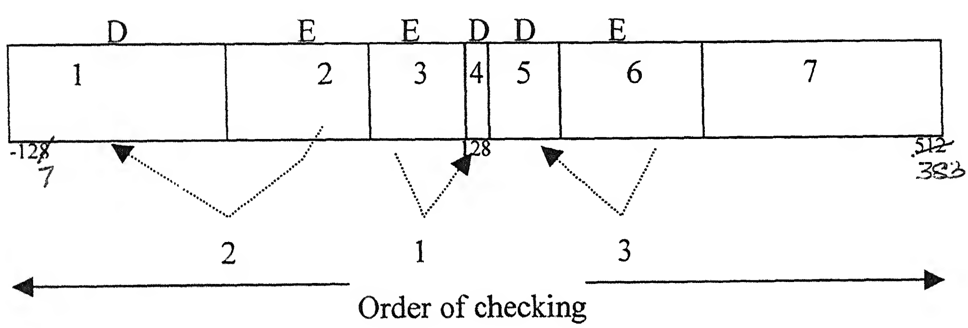
Adjustment Algorithm -I for 7 range quantization

2.5 Different Adjustment Algorithms

There exist different adjustment algorithms that can achieve the adjustment process with different performance levels. Two of such algorithms are discussed here of which adjustment algorithm-II performs more efficiently than algorithm-I.

Algorithm-I:

It checks for the deficit range from smoothest range through contrast range i.e. in the order range 4, range 3/range 5, range 2/range 6, range 1/range 7. While attending smoother ranges of equidistant from smoothest range (index 4), it prefers range with larger size. For instances if index 3 and index 5 are found deficit, priority is given to index larger size.



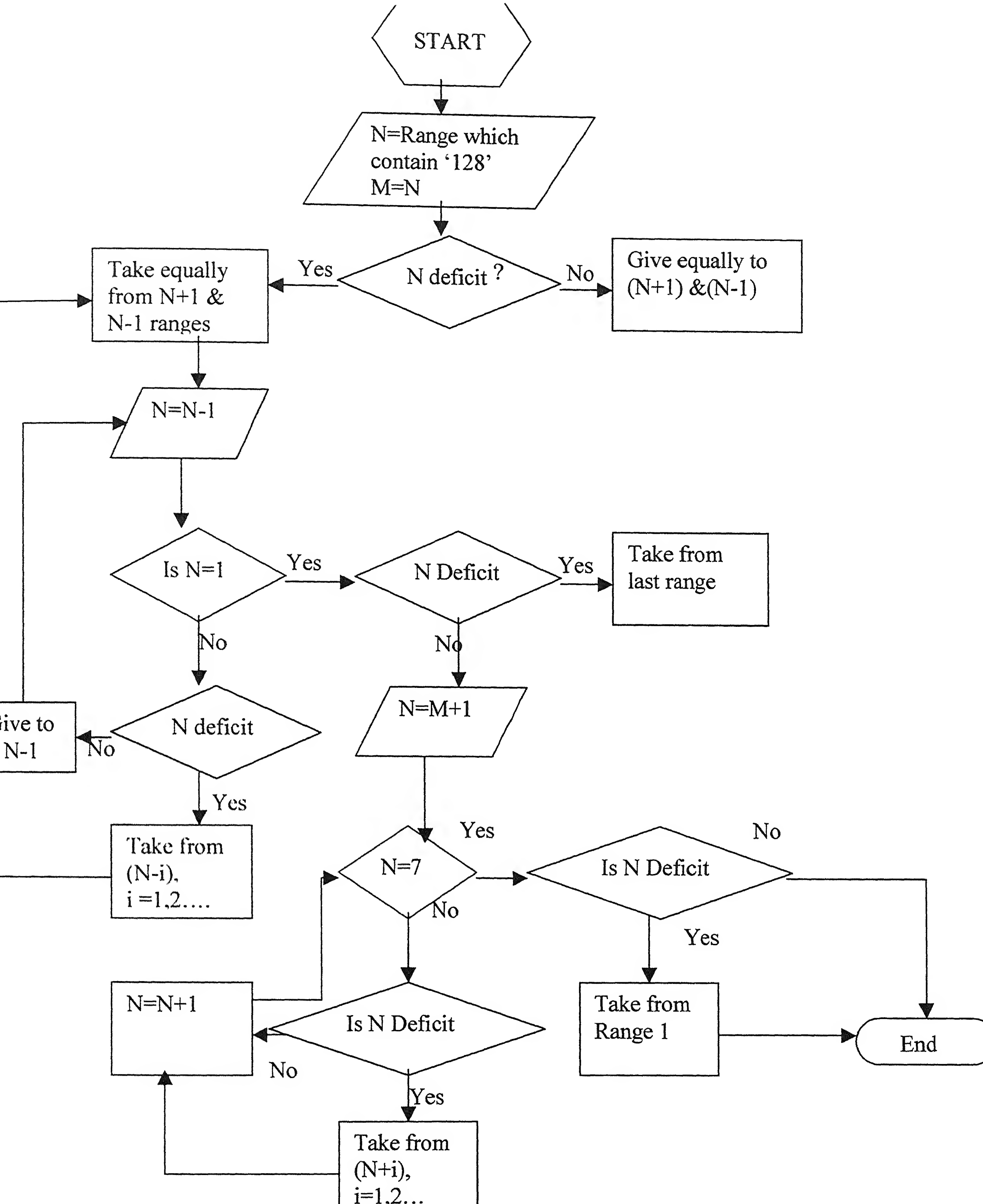
The main requirement for grey value adjustment arises due to occurrence of deficit ranges, which are resulting choice of range boundaries.

The aim of any of algorithms should be to select pixels from excessive range for minimum distance transfer to deficit range and at the same time one has to note that pixel transfer for high-contrast deficit range can tolerate more distance transfer in comparison to that for smoother range.

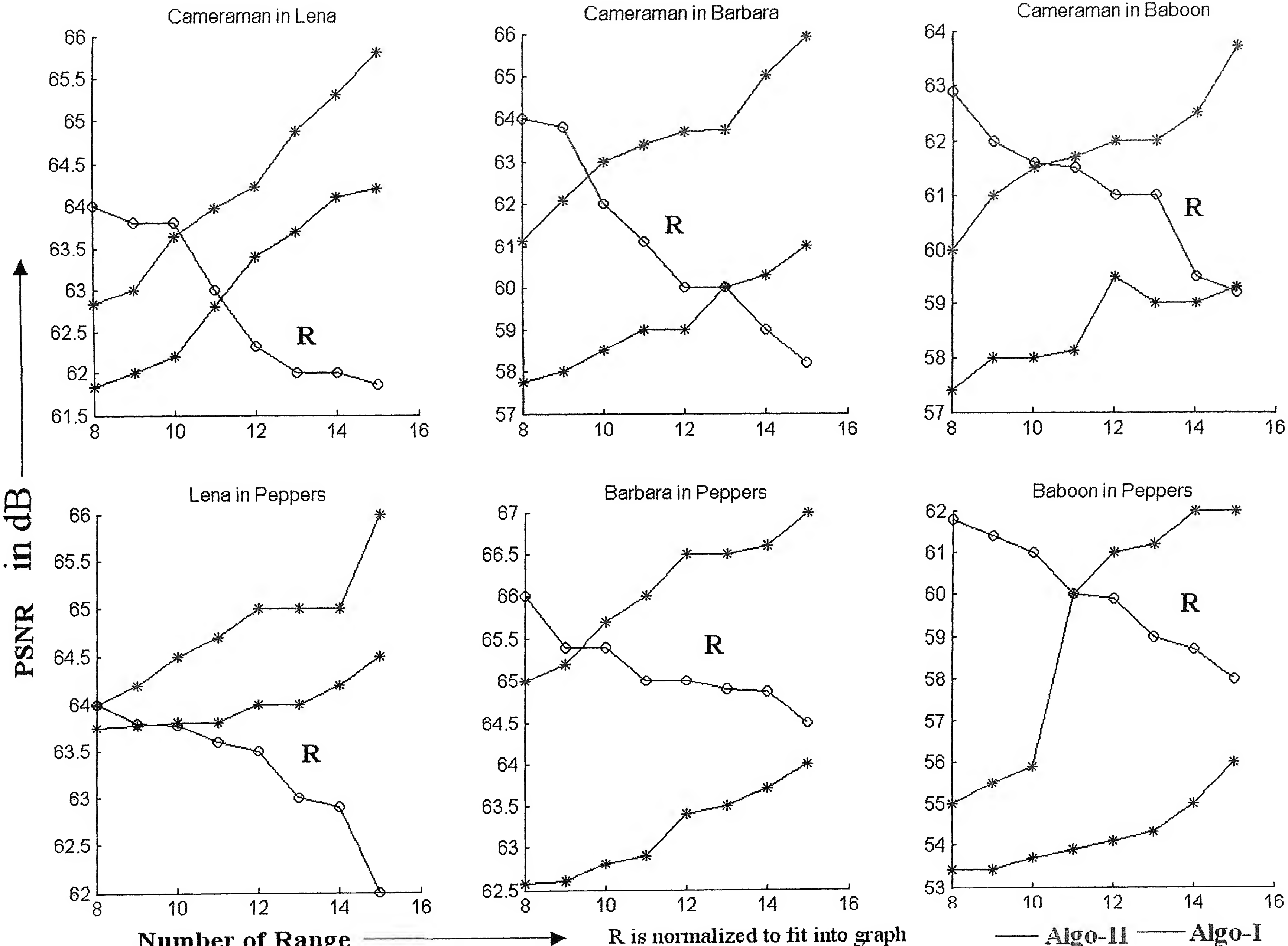
Algorithm 2:

This is an entirely different algorithm from the earlier one. In this algorithm, each excess range supplies pixels to adjacent deficit ranges, and collects (if in deficit) pixels from adjacent ranges, irrespective of their status (i.e. excess or deficit). It starts from the middle (smoothest) range; if found deficit, it takes pixels of lowest (highest) grey values available in the immediate right (left) adjacent range, even if these donors themselves might be in deficit. In the process, when processed towards left direction from middle, range '1' (the extreme left range) may become deficit (excess), in which case it collects from (donates to) the last range (the extreme right range) which is an equally contrastive range. Similarly, when processed from the middle rightwards, the last range may become deficit (excess), and it would take pixels from (gives pixels to) the first range. So whenever transfer is from the right side range, it is of the smallest available grey values in the right range. Similarly when transfer is from left side range, is is of highest available grey values. In this approach, some grey values may appear in more than one range. This new approach, is most suitable when the size of S_d and C_d (that considered for hiding image only, not leading information) are same, in which case if last range is excessive, (it indicates that range '1' is in deficit status) it will supply all its excess to range '1'.

This new proposed algorithm becomes less effective as size of C_d (that considered for hiding image only, not leading information) increases relative to size of S_d., as it would result in increased size of edge ranges i.e. range'1' and last range, to be very high than that of S_d in which case there would be heavy replacement of smoother areas of cover image with contrastive of secret image. The remedy for this situation is a slight change of the above algorithm as follows: Starting from middle range proceeding to either of ranges (one after other as above), whenever excess range encountered, starting from smoothest range it should keep the excess grey values with itself. Whenever deficit range is faced, take equal number from its either side excess ranges.



Flow Chart for Adjustment algorithm-II



Cameraman in Lena
Cameraman in Barbara
Cameraman in Baboon
Lena in Peppers
Barbara in Peppers
Baboon in Peppers
R
PSNR in dB
Number of Range
R is normalized to fit into graph
Algo-II
Algo-I

Chapter 3

Image Embedding and Inverse Image Differencing

3.1 Grey Value Replacement:

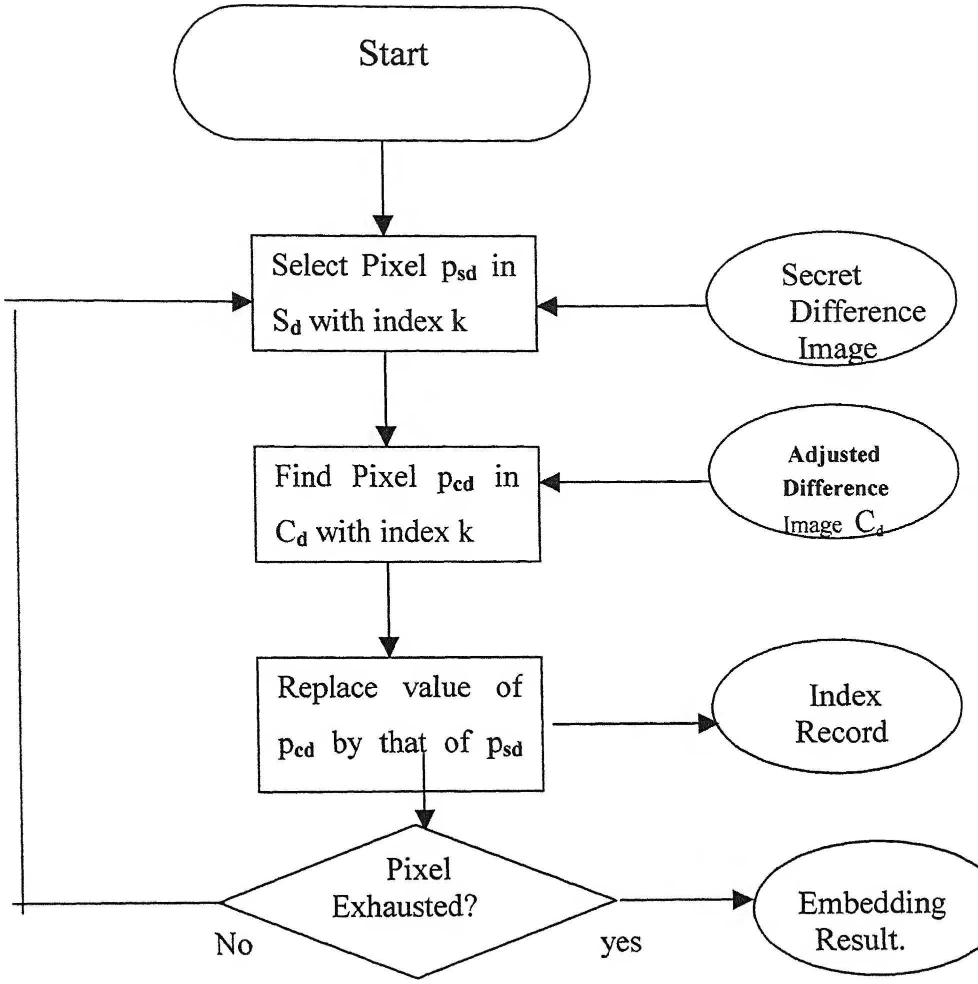
After all insufficient ranges are identified and the numbers of pixels to be changed are accumulated, the work of adjustment is performed by randomly traversing through C_d, visiting each pixel only once. For every visited pixel in C_d with a grey value, that needs to be changed, we change it to a (any) value of insufficient range according to one of the algorithms for accumulation mentioned above. (In fact we don't even need to change the grey values, but just make a record of those pixel locations for replacement work (which would result in same effect as that of actually changing grey values). Since the adjustment work may cause perceptible changes in some pixels, the random working mechanism provides a way to scatter the distortion over the whole image, rather than concentrating it in any particular region alone. After the adjustment process, the cover difference image is now ready for use in the subsequent step, that of *grey value replacement*.

Now, we can replace a pixel in C_d having a certain grey value (of a certain index) with a pixel of the same index in S_d. Likewise all pixel grey values of S_d can be placed into C_d. For the purpose of easily finding the desired pixel P_c in C_d, which has the same index as P_s, we rearrange all the pixels into an n-list structure. Every list in the n-list structure has an index, which corresponds to the index of grey-value range mentioned previously. Every entry in the list is a record of a location in C_d that has that index. A process of traversing each pixel of C is applied, in which we put the visited pixel location at the rear of the corresponding list according to the index of its grey value. For example if the location of C_d visited, 8605 is of grey value 128, and 128 belongs to the range whose index is 'k', the we add a node with a value 8605 to the rear of list 'k'.

Moreover if we permute the traversing order in C_d, using a pseudorandom number generator, instead of scanning sequentially, it serves to achieve cryptography. If an illicit user does not have the seed of the pseudorandom number sequence, he cannot find out the correct traversing order and fails to extract the hidden image.

After constructing the list structure, we find for every pixel P_s in S_d, the corresponding pixel in the list structure that has the same index to accomplish the replacement. We process every pixel in S_d in the zigzag order mentioned earlier. For each visited pixel P_s in S_d of index 'k', we extract the head element in k^{th} list of the list structure which denotes the location in C_d and replace the grey value of this location with that of P_s. For example, if we visit a pixel in S_d with a grey value 120 and index 'k' then we extract the head element from list 'k', whose value 8605 is a location in C_d with grey value 128 where we want to do the replacement work. The grey value of the visited pixel is used to replace that appearing in location 8605 of C_d, i.e. 128 is changed to 120. Next time when same range (k) pixel is visited in S_d, it replaces grey value of the location in C_d given by the second head element from list 'k', as the original head element has been moved to the bottom of the list.

The replacement value is finished after all pixels in S_d have been processed. The grey value replacement value is efficient because there is a one-to-one mapping between the pixels of the S_d and C_d, which resulted from traversing through S_d and C_d only once.



Algorithm for Grey Value Replacement

3.2 Embedding of leading Information:

Now the embedded cover difference image is embedded with secret difference image. To recover the secret difference image the receiver requires to know the following:

1. The Pseudorandom number sequence.
2. The index table, which was constructed during the replacement process.
3. Size of the secret image.
4. The number of quantized ranges, size of each range (required if Huffman coding is used) and boundaries of each range.

Of the above, the pseudorandom number sequence, which is fixed, need not be sent each time and is grafted only once at the receiver. The remaining information that provides a way to send different secret images of different sizes with different range boundaries has to be sent each time with the respective stego image. As the indices with the values 1 to n, the number of ranges, concentrate on just a few values, it helps to use Huffman coding method to reduce the number of bits required. The entire leading information taken as a bit stream is embedded in 3 bits of the LSBs of the cover image pixels, or 6 bits of the stego difference image pixels which were unused for embedding the secret difference image. This can be done by traversing randomly through the cover difference image, without touching the embedded pixels.

Instead of using a fixed number of 3 bits of LSBs of the cover image for embedding the leading information, one can use variable length coding. In VLC method one can use more bits in the contrastive areas.

3.3 Inverse image differencing

In the inverse image differencing process, which produces the stego image, for each difference value $T_d(i)$ in the embedded cover difference image (the so-called stego difference image) T_d ($i = 1,2,3,\ldots$No.of pixels in T_d), an inverse calculation is performed upon each consecutive pixel pair $T_d(2*i-1)$, $T_d(2*i)$ to find the corresponding grey

3.2 Embedding of leading Information:

Now the embedded cover difference image is embedded with secret difference image. To recover the secret difference image the receiver requires to know the following:

1. The Pseudorandom number sequence.
2. The index table, which was constructed during the replacement process.
3. Size of the secret image.
4. The number of quantized ranges, size of each range (required if Huffman coding is used) and boundaries of each range.

Of the above, the pseudorandom number sequence, which is fixed, need not be sent each time and is grafted only once at the receiver. The remaining information that provides a way to send different secret images of different sizes with different range boundaries has to be sent each time with the respective stego image. As the indices with the values 1 to n, the number of ranges, concentrate on just a few values, it helps to use Huffman coding method to reduce the number of bits required. The entire leading information taken as a bit stream is embedded in 3 bits of the LSBs of the cover image pixels, or 6 bits of the stego difference image pixels which were unused for embedding the secret difference image. This can be done by traversing randomly through the cover difference image, without touching the embedded pixels.

Instead of using a fixed number of 3 bits of LSBs of the cover image for embedding the leading information, one can use variable length coding. In VLC method one can use more bits in the contrastive areas.

3.3 Inverse image differencing

In the inverse image differencing process, which produces the stego image, for each difference value $T_d(i)$ in the embedded cover difference image (the so-called stego difference image) T_d ($i = 1,2,3,\ldots$No.of pixels in T_d), an inverse calculation is performed upon each consecutive pixel pair $T_d(2*i-1)$, $T_d(2*i)$ to find the corresponding grey

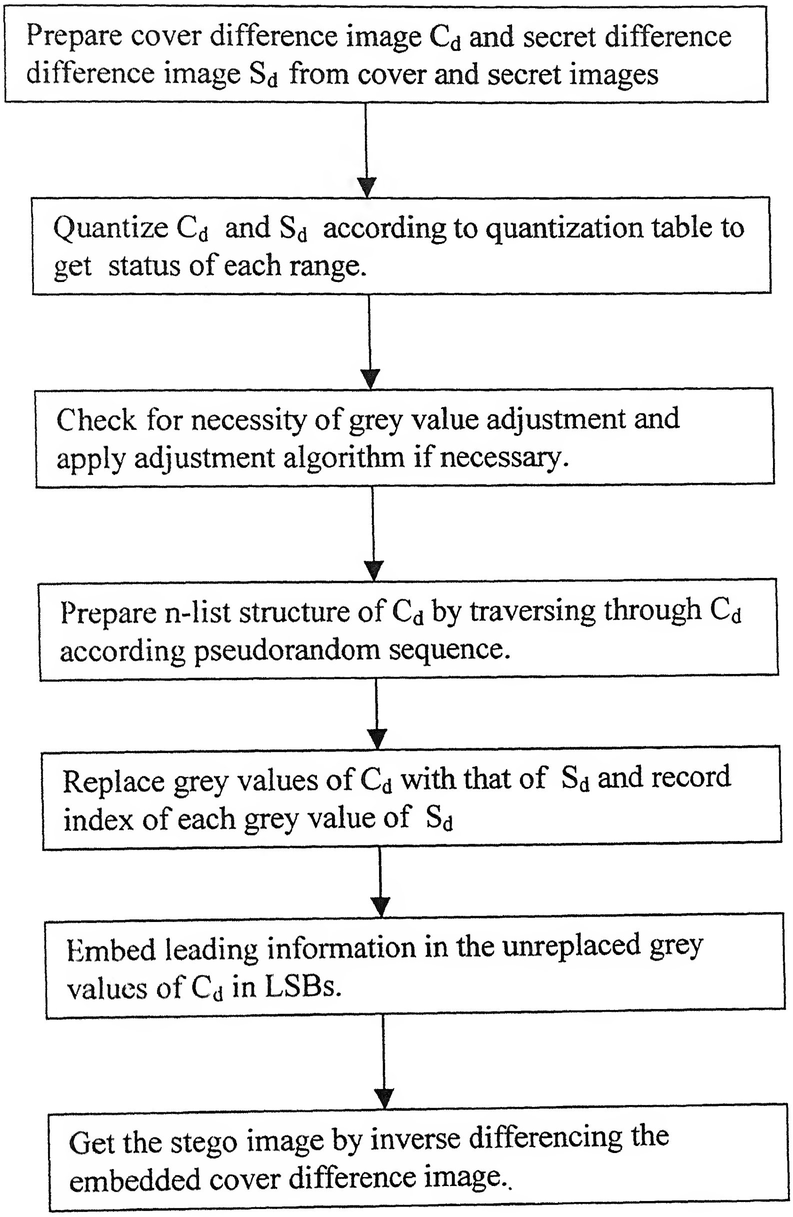
values $T(2*i-1)$, $T(2*i)$ of the actual stego image, T. We know that the difference of $T(2*i-1)$ and $T(2*i)$ in the stego image is $T_d(i)$. To cause the least perceptual distortion, information about the grey values ($C(2*i-1)$, $C(2*i)$) of the corresponding two-pixel subimage in the original cover image, C is needed. We produce ($T(2*i-1)$, $T(2*i)$) according to the following equations:

$$T(2*i) = C(2*i) + [T_d(i) - C_d(i)]/2$$

$$T(2*i-1) = C(2*i-1) - [T_d(i) - C_d(i)]/2$$

The above equations together satisfy the requirement that the difference of $T(2*i-1)$ and $T(2*i)$ is $T_d(i)$. Because these equations cause changes in $C(2*i-1)$ and $C(2*i)$ nearly equally to produce $T(2*i-1)$ and $T(2*i)$, the distortion caused by embedding into $C(2*i-1)$ and $C(2*i)$, is averaged over this two-pixel pair ($T(2*i-1)$, $T(2*i)$) and so is less. Some of the calculations may cause $T(2*i-1)$ and $T(2*i)$ to fall off the boundary of the range [0,255] of a pixel value. In such a case, set the pixel value to the boundary value, i.e. to 0 or 255 as applicable, and readjust the other pixel value to a new value to preserve the difference value of $T(2*i-1)$ and $T(2*i)$ is $T_d(i)$.

So the information is in the difference of the two-pixel pair. If noise affects this pair, that particular difference value changes and if noise affects equally the two pixels of the pair, then the effect is nil. This is the main advantage of adopting such differencing method for the cover image. On the other hand if we had adopted on the cover image a differencing method similar to used in the secret image, then *all* successive pixels values will be distorted even if only one pixel gets affected by noise (Cumulative distortion).



Algorithm for hiding the secret image in the cover image

Chapter 4

Extracting the Hidden image

4.1 Recovering secret image

The steps involved in the process of extracting the secret image are as follows:

1. Make the stego difference image from the stego in exactly the same way the cover difference image was prepared from cover image. (as discussed in section 2.2)

2. Extract the leading information, i.e. the size of the secret image, number of ranges used for quantization, boundaries of each range and the table of indices from the stego image using the pseudorandom sequence (assumed to be known to the receiver).
3. Using the seed of the pseudorandom sequence, traverse through the stego difference image and prepare an n-list structure of locations according to range boundaries, by arranging the locations of pixels traced according to the pseudorandom sequence in respective list. It is to be noted that this n-list structure is exactly same as that prepared before embedding the secret image, as the pseudorandom sequence followed here is same as that while embedding. That is why we get an *exact* recovery of the secret image from the stego. If at any stage, stego image has undergone modification like cropping, filtering, lossy compression, image enhancement, etc., the list structures prepared will differ from that of embedding stage, which produce noise (horizontal or vertical stripes) in

the recovered secret image. Thus most of the spatial domain image hiding techniques are more sensitive to image transformations.

4. According to the index record, pick the grey values from the locations given by the n-list structure and place them in order in the same zig-zag manner to prepare the secret difference image. For example, if index of the 1^{st} pixel is 'k', then pick grey value of stego difference image from the location given by the head element of list 'k'.

After all pixels, according to the index record, are collected, the reconstructed secret difference image is further subjected to inverse differencing (of the kind meant for S_d) to prepare the secret image S' by using the following equation:

$$S'(i)= S'(i-1)+ S_d'(i)-128$$

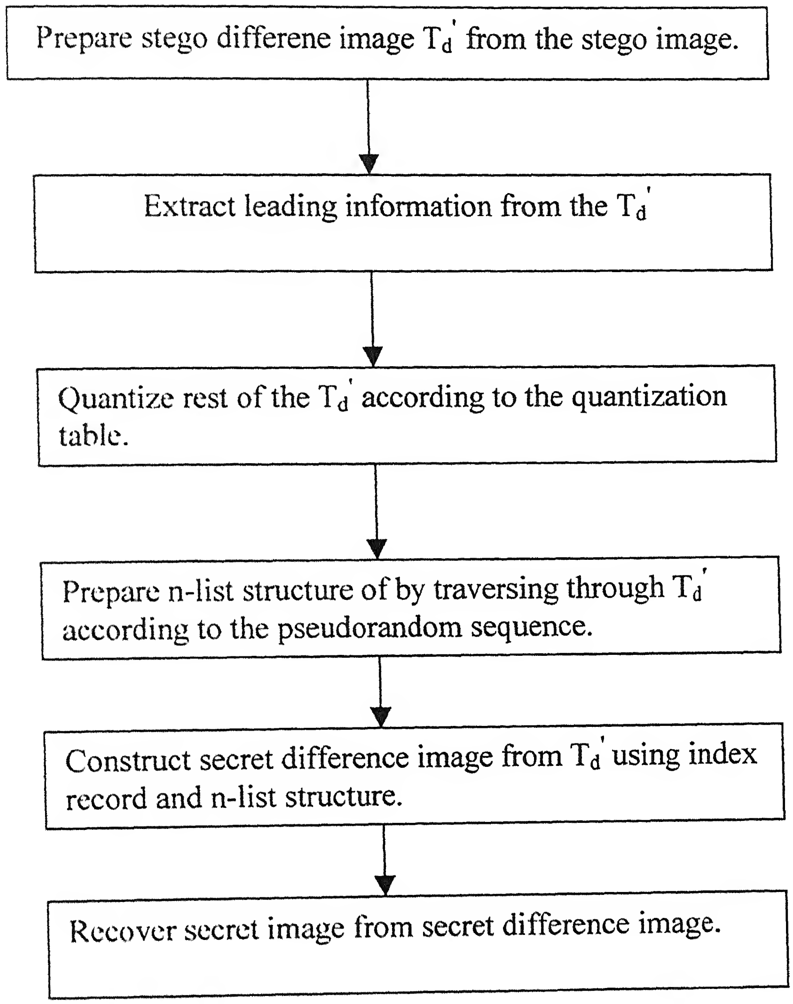
where S': Reconstructed secret image

S_d': Reconstructed secret difference image

$S'(1) = S(1);$

4.2 Color images

Color images can be visualized as R, G and B components in RGB-space or as Y, C_b and C_r components in Y C_b C_r -space. All these individual components can be treated separately for the purpose of steganography. Thus R, G and B components of a secret image are embedded into R, G and B components of the cover image respectively. In this thesis we have not exploited any of color perceptual properties relating to HVS, which can help us to increase the embedding capacity. Above everything else, one may be reasonably expect a high degree of correlation between the colour components of any image, which we might exploit to create the index ranges for the 3 components jointly, with almost a third of the expected effort.



Algorithm for reconstruction of secret image from stego image

4.3 Image sequences

Image sequences can also be embedded into other sequences, considering each image as a frame. In this, temporal redundancy between successive frames can be exploited besides spatial redundancy between successive pixels of a frame, to help us achieve steganography. Temporal correlation between successive frames of any well-sampled video sequence (and this applies to both the cover and the secret sequence) will significantly reduce the computations that would seem to be required, at first glance.

CHAPTER 5

Results, Conclusions & Scope for Future Work

5.1 Results & Conclusions

In this project, secret images of size 256 X 256 and cover images of size 512 X 512 were taken for conducting experiments. We have used adjustment algorithm-I and proposed adjustment algorithm-II for all experiments. It was observed that while hiding secret images of high contrast (baboon) i.e. high standard deviation in low contrastive cover images, heavy grey value replacement is taking place resulting in comparatively low PSNR. Also observed is that, reduction in cost factor 'R' keeping total number of deficit pixels same, the PSNR is increased, while exercising different range boundaries and different range numbers. Even the lowest PSNR stego images are with good quality, as most of data is hidden the contrastive/edge areas. Using this differencing method, one can achieve exact recovery of the secret image. Even two secret images can be embedded with very less noticeable changes in the
cover image, without leading information, that is to say, the maximum embedding capacity is 50% of the cover image. The adjustment algorithm-II has shown tremendous improvement of 3 dB in PSNR over algorithm-I. This algorithm is more effective when sizes of S_d and C_d are same.

While hiding the color images, the poor PSNR of color stego as that of B/W stego is noticed with high imperception (eventhough degradation is same in color and B/W stego images) due to sensitivity variation of HVS to different colors.

For hiding sequences of images we have taken standard image sequence TT game and Claire, the news reader. The main problem with image sequences considered for steganography is the time taken for computation as each image is hidden into its respective cover image.

For hiding one grey image into the cover, it is taking 2.5 min. at the maximum and 1.5 min. to recover the secret image from stego. (In MATLAB)

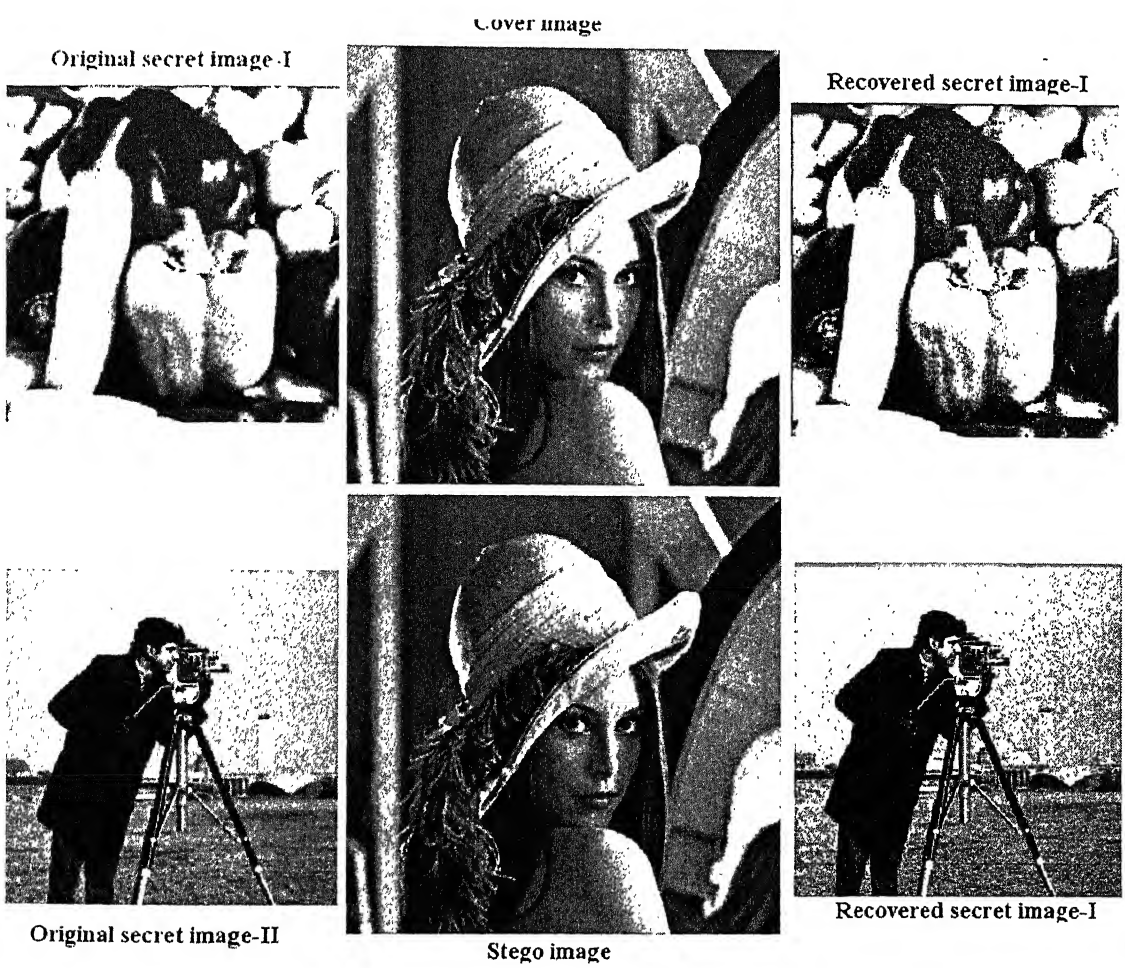


Fig 5.Cover image (512 X 512) hiding two secret images (256 X256)

Original Cover Image

Stego Image

Fig 3. Lena embedding cameraman & Peppers (Zoomed In)

original Image

Recovered Image

Cover Image

Stego Image

original cover image

stego image

Lena hiding Baboon using adjustment algorithm-I(Zoomed In)

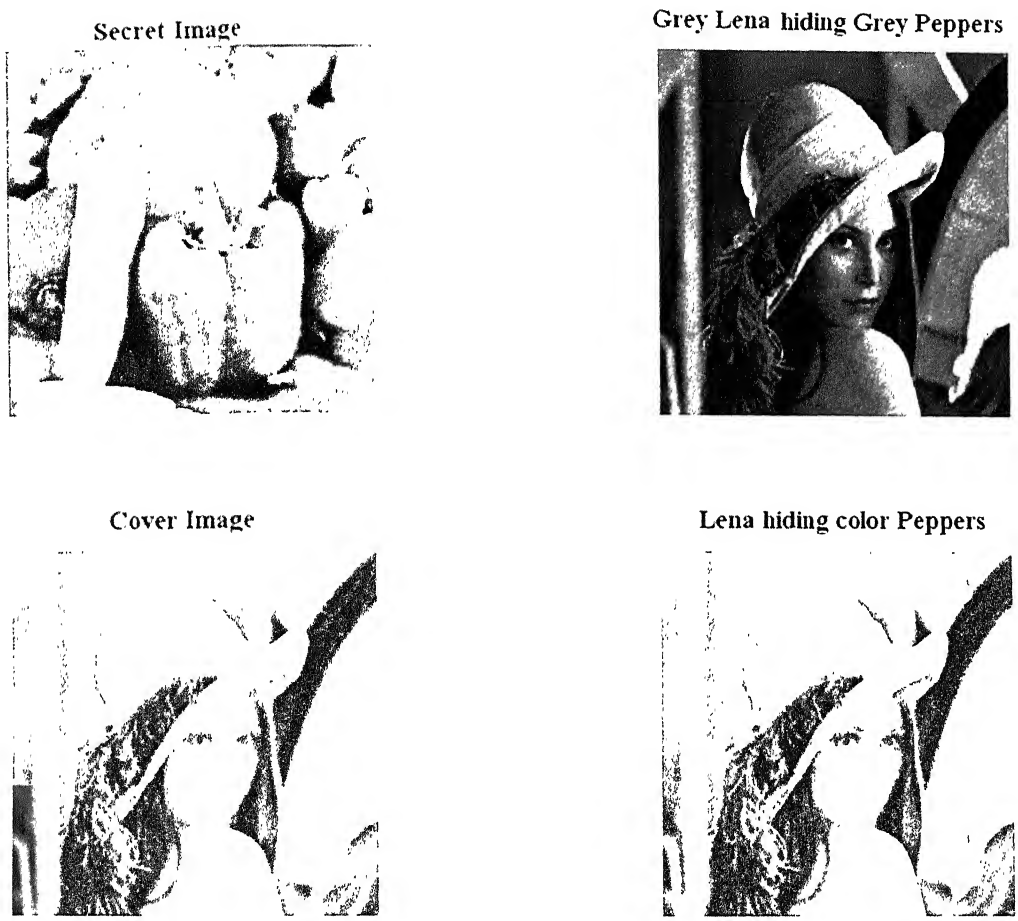
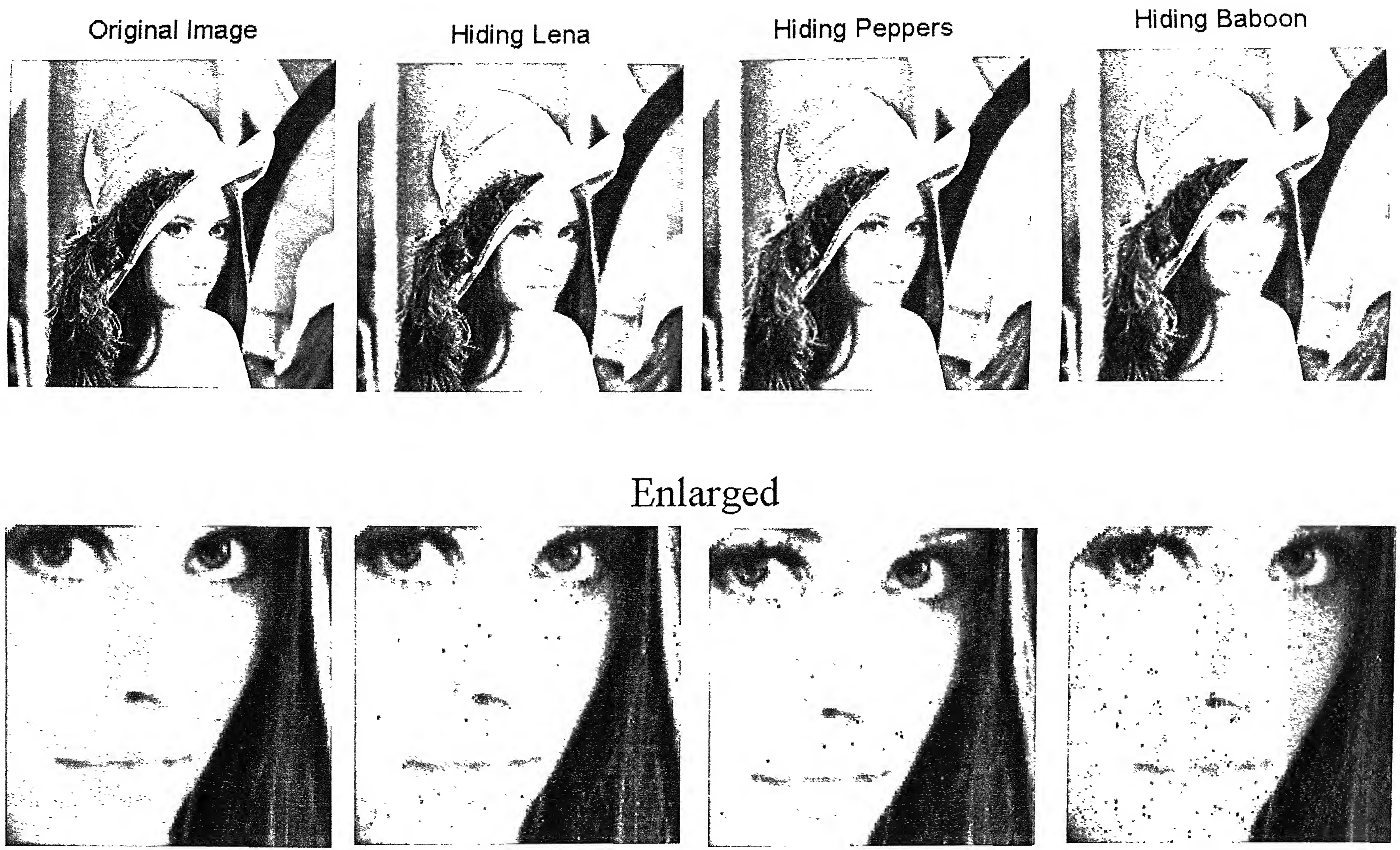
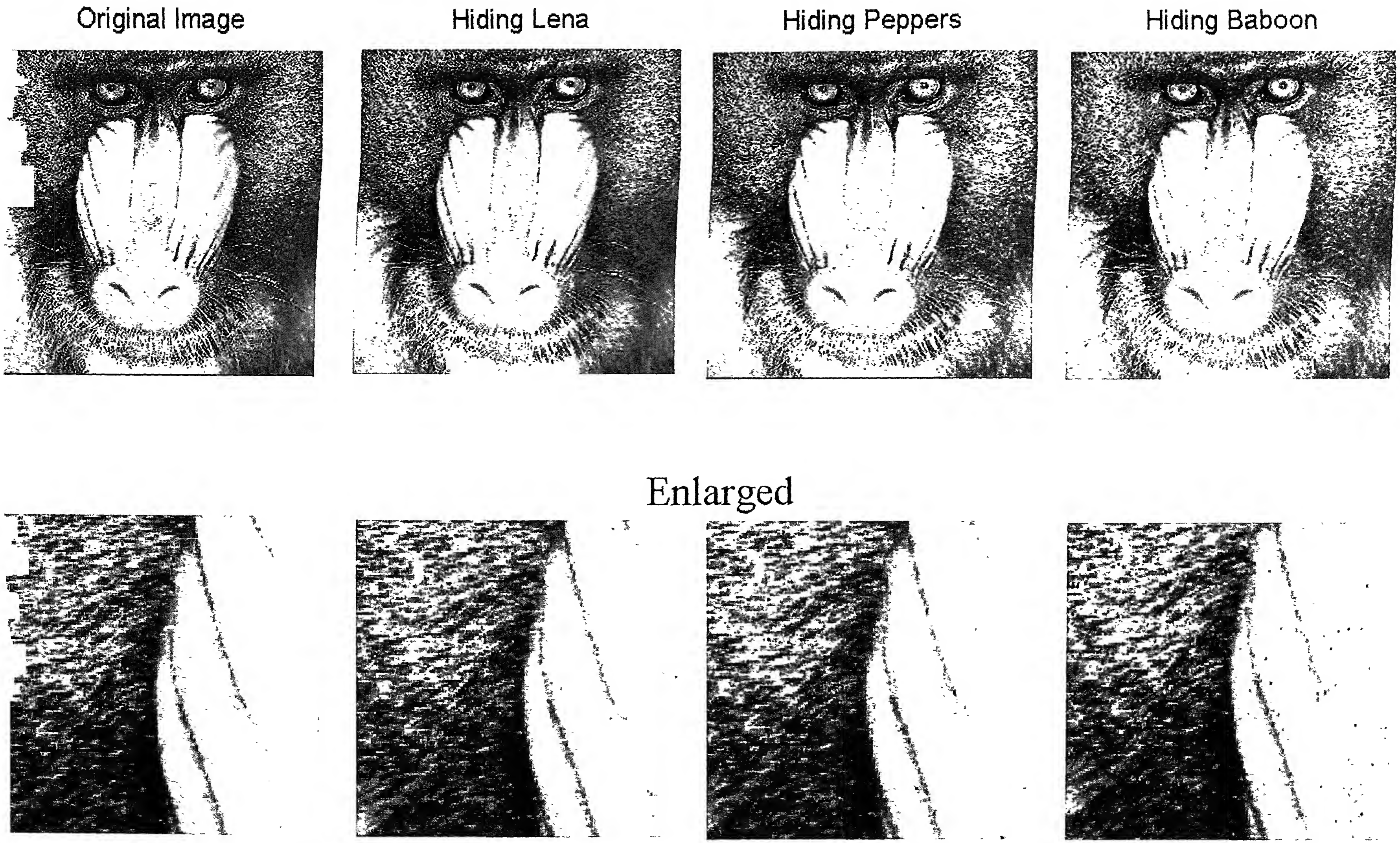


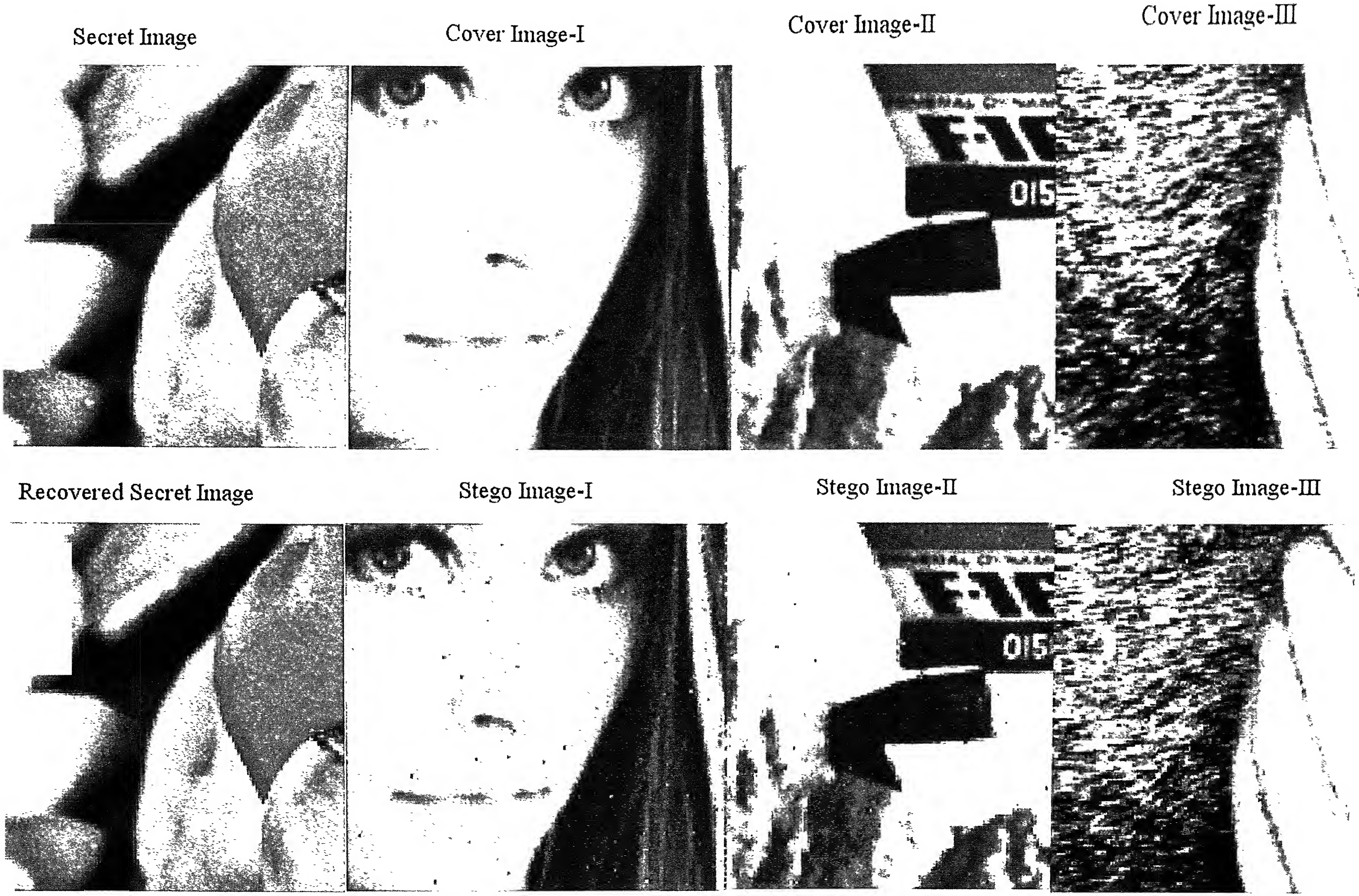
Fig 7 Depicting Low PSNR of Stego hiding grey images



Lena image hiding different secret images with leading information and using adjustment algorithm-I



Baboon hiding different secret images with leading information(Algo-I)



Different cover images hiding same secret image-Peppers

5.2 Suggestions for Future Work

To increase the embedding capacity, all kinds of practices like LSB method, frequency domain methods, spatial domain methods, (including image differencing method), etc can be done on the secret image and this stego image can be subjected to second order embedding in yet another (second order cover) image. If images or sequence of images need to be sent over a broadcasting channel, it is possible without affecting the quality of the channel.

Still better adjustment algorithms can be thought of to achieve better PSNR. While doing color steganography, if color perceptual error measures are considered, better PSNR can be achieved. Also while performing video steganography, if temporal redundancy is used, much time and effort can be reduced.

Bibliography

1. D.C.Wu & W.H.Tsai. '*Spatial-domain image hiding using image differencing*', IEE Proc. Vis. Image Signal processing, vol 147, No.1 Febraury, 2001.
2. A. K. Jain. '*Fundamentals of Digital Image Processing*'. Prentice Hall, Englewood Cliffs, NJ, USA, 1989.
3. Y.K.Lee & L.H.Chen. '*High capacity image steganographic model*', IEE Proc.Vis. Image Signal Processing vol. 147, No.3, June,2000.
4. Ross J. Anderson & Fabien A.P. Petitcolas. '*On the limits of Steganography*', IEEE J.Sel.Areas Commum., vol.16,No.4, May,1998.
5. Brassil, J.T., LOW, S., Maxemchuk, N.F., and O' Gorman, L. '*Electronic marking and identification technique to discourage document copying*', IEEE J.Sel.Areas Commum., 1995,13,(8),pp.1495-1503.
6. Koche,E., and Zhao, J. '*Towards robust and hidden image copyright labeling*'. Proc. of IEEE workshop on Nonlinear signal and image processing, 1995, Thessaloniki, Greece, pp.452-455
7. Liaw, M.S., & Chen, L.H.: '*An effective data hiding method*'. Proceedings of IPPR conference on 'Computer vision, graphics, and image processing', 1997, Taiwan, R.O.C., pp.146-153.
8. Pfitzmann, B.: '*Information hiding terminology*'. Proceedings of first int'l workshop on 'Information hiding' Lecture Notes in Computer Science' No.1174, (Springer-Verlag, Berlin, 1996), pp. 347-349.
9. Hsu,C.T., & Wu, J.L. '*DCT-Based watermarking for video*' IEEE Trans. Consum.Electron.,1998,44,pp.206-216
10. *http://imaging.comp.glam.ac.uk/testimages.htm* for standard test images.